第1章

複雑化するシステムの信頼性確保には冗長設計が欠かせない



冗長設計の位置付けから具体的な手法まで

檜原弘樹

ここでは，機器の信頼性を高める手段の一つ，冗長設計について，その位置付けから設計の取り組み方までを解説する．冗長設計においては，設計者自身が担当する回路だけでなく，システム全体を理解し，どの部分を冗長構成とすることが最善なのかを見極めるバランス感覚も大切である．（編集部）

高信頼性システムを実現するためには，特集1で述べられているように，さまざまな設計解析手法や設計技術が使われます．本稿では，対策の一つとして用いられる冗長設計の枠組みについて説明します．

1 冗長設計の位置付け

冗長設計を行うことにより，どのような効果が見込まれるのでしょうか．高信頼性を要求されるシステムには，故障や誤動作を起こさないように，設計余裕を持たせる必要があります(**図1**)．

ハードウェアを構成する部品や材料のばらつきを考慮した設計余裕を持たせるとともに，システムが使われる環境や，装置が耐えられる環境(温度範囲，湿度範囲，気圧範囲，振動・衝撃レベル，放射線レベルなど)も考慮する必要があります．

● ばらつきへの配慮はソフトウェア設計においても必要

製造時のばらつきについても考慮する必要があります．これはハードウェアの設計ばかりでなく，ソフトウェアの設計にも当てはまります．例えば，デバイスや製造時の調整のばらつきがソフトウェアに対する割り込みのタイミングに影響を与えたりすることもあります．

万が一，故障が検出されたり誤動作が発生したりした場合でも，破局的な故障には至らず，最悪でもいったん機器の動作を中断し，マニュアル操作によって運転や復帰ができるようなフェイルセーフ(fail-safe)機能が必要とされます．

● 耐故障への要求は高まるばかり

近年，組み込みシステムの提供する機能やサービスの高度化(インテリジェント化)が進んでいます．マイクロプロセッサを内蔵するばかりでなく，さまざまなセンサや通信機能が組み込まれ，個々のセンサ出力やネットワーク経由で送信された情報を有機的に統合して運用する自動化・自律化機能なども求められるようになってきました．

組み込みシステムに用いられるマイコンの高度化により，このようなシステムはますます増えるものと考えられます．これらの機器は，オペレータの操作を必要とするものばかりでなく，一度システムが起動されると以降は長時間無人で稼働する機器も増えており，無人で運用する組み込みシステムへの信頼性要求も高度化しています．さらに，航空機やロボットの自動操縦システムのように，組み込みシステムがオペレータと協調的に動作するような高度な自動化技術も求められるようになってきています．

● 単一故障点を除去するのが冗長設計

このような組み込みシステムでは，フェイルセーフからさらに進み，異常や故障が生じても適切な動作を継続する

KeyWord
フェイルセーフ，fail-safe，フェイル・オペラティブ，fail operative，耐故障性，故障ケース，
クロス・ストラップ，コールド・スタンバイ方式，待機冗長方式，ホット・スタンバイ方式，常用冗長方式

フェイル・オペラティブ(fail operative)機能が求められるようになってきています．特に，安全性が求められるシステムに実装するフェイル・オペラティブ機能は，信頼性の高いものでなければなりません．一つ故障が発生しても継続して稼働できるものを1フェイル・オペラティブと言います．人命にかかわるようなシステムや，社会インフラにかかわるようなシステムでは，二つ故障が発生しても破局的な故障には至らず，最低でも正常動作に復帰できる2フェイルセーフであることが求められることが少なくありません[注1]．

組み込みシステムへの耐故障性の要求はますます高度化してきており，フェイルセーフ，さらにフェイル・オペラティブ機能を実現するためのさまざまなアーキテクチャが開発されています．これらのアーキテクチャの一つとして用いられるのが，冗長構成です．ある特定の部分の故障によりシステムの機能が失われてしまうとき，その部分を単一故障点(single point of failure)と呼びます．システムから単一故障点を除去するための手段として冗長設計が用いられます(**図2**)．

2 さまざまな立場から見た冗長設計がある

冗長設計は，組み込みシステムが持つ演算機能や，故障検出機能，故障分離機能，再構成機能，再同期機能をそれぞれ分散化することにより行われます．

● 各機能を分けて考えることから始まる

演算機能を冗長化する方式としては，多数決方式や，時間軸上の冗長設計として同じ演算を2回以上実行して結果を確認する方式などがあります．

故障検出機能を冗長化する方式としては，誤り検出・訂正符号や，複数のプロセッサがプロセッサそのものの出力とほかのプロセッサの出力とを比較する方式などがあります．

故障分離機能を実現するためには，故障が生じた部分を物理的，あるいは論理的に隔離できる必要があります．システムがうまく階層的に構成されている場合には，故障をマスクしてほかに波及させずに済ますこともできます．故障した部分が自発的にシステムから離脱できるようにすれば，故障分離機能も分散化した冗長設計を行えます．このようなシステムをセルフ・パージング・システムと呼びます．



図1　冗長構成の例1
予定時刻に起きるための準備を念入りに行った．

故障分離機能が実現できていれば，冗長系に切り替えたり，故障した部分を切り離したりするなどして，失われた機能をほかの機能で肩代わりさせるようなシステムに発展させることも可能です．これらは再構成機能と呼ばれますが，完全に元の機能・性能を維持せずに，一部の機能を縮退させたり，性能を落としたりすることによりシステムの運用を継続することを優先させる場合もあります．

システムに故障が発生する以前の状態が保持されていれば，システムを再構成した後に，その時点までいったん後戻りして処理を継続させることも可能です．このような再同期機能が組み込まれていれば，速やかに通常の処理サイクルに復帰させることが可能になります．

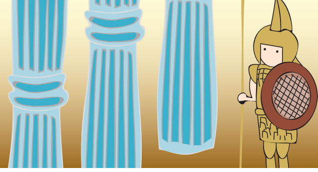
● 故障の解析を徹底し，冗長設計の方針を決める

これらの冗長設計を行う際は，十分に故障ケースの解析を行い，一過性の故障に対応するのか，永久故障に対応するのかなども考慮して最適な方式を選択する必要があります．

例えば，人工衛星に搭載されるコンピュータについて見てみましょう[(1)]．宇宙機器搭載用のオンボード・コンピュータ(onboard computer：OBC)は，通常の組み込みシステム用計算機と比較すると，その使われる環境において多くの点を考慮しなければなりません．外部環境としては，宇宙空間特有の温度差や放射線環境を考慮する必要があります．温度差については，衛星システムの熱制御技術の進歩により，運用中の衛星内部の温度変化を非常に緩やかにすることが可能となっています．しかし，輸送，打ち上げ

注1：技術的に簡単な方から並べると，1フェイルセーフ(一つ故障が発生しても正常動作に復帰できる)，1フェイル・オペラティブ，2フェイルセーフ，2フェイル・オペラティブ(二つ故障が発生しても継続して稼働できる)となる．

1



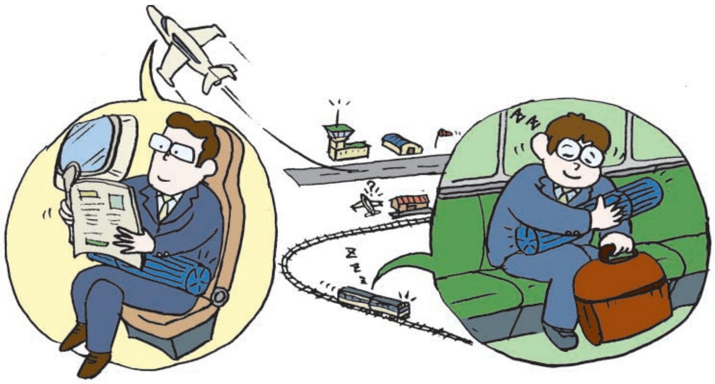


図2
冗長構成の例2
結局，僕って信用されていなかったのね．

なども含めたライフサイクル全体を考慮した場合，組み込み機器単体として動作を保証する温度幅は依然として60 ℃以上にもなります．

また，宇宙空間では放射線レベルが高く，惑星の近傍や太陽風などに含まれるγ線，および銀河宇宙線などに含まれる重粒子イオンなどが存在します．前者は主として，その総線量によるデバイス特性の劣化を招き，後者はソフトエラー（データの反転など）またはラッチアップなどの過渡不良を引き起こします．こういった放射線の影響に対して耐性を持たせる必要があり，さまざまな冗長設計が用いられます．

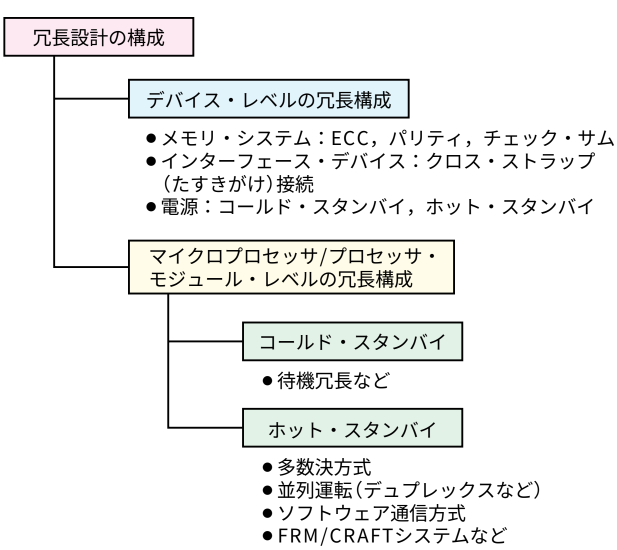


図3　冗長設計の構成
冗長設計を構成するには，デバイス・レベルで冗長を構成するものと，マイクロプロセッサやプロセッサ・モジュール・レベルで冗長を構成するものがある．ハードウェアの機能で冗長化するばかりでなく，ソフトウェアを用いて冗長化する方法もある．

3　冗長設計は人命とお金のトレードオフ？

システムの信頼性を高める方法を検討するにあたり，「信頼度を定量的に検討する方式」と，「システム的に冗長系を構成する方式」が存在します．

前者の方式には，米国の宇宙機システムや欧州の鉄道規格などがあります．これらのシステムでは，システムの稼働率や人命にかかわるリスクも数値的に管理します．

一方，システム的に代替機能を活用して機能を継続する考え方もあります．例えば，小惑星「イトカワ」に世界で初めて自動着陸，離脱に成功した日本の人工衛星「はやぶさ」では，スラスタやジャイロが故障しても，イオン・エンジンで制御を続けています(2)．この考え方は「機能冗長」と呼ばれ，信頼度の定量的な表現ばかりでなく，さまざまな運用ケースが検討されます．

日本市場では，信頼度とコストとのトレードオフはあまりなじまず，ある期間中に故障しないことが暗黙のシステム要求となることが少なくありません．一方，欧米の数値的な信頼度に基づくトレードオフの考え方から，高い信頼性を持つ部品の製造方法がシステマチックに定義されるなどの利点があり，これら二つの考え方は今後も相補的に用いられていくものと考えられます．

4　冗長設計への具体的な取り組み方

これまで述べたように，冗長設計は単一故障点の除去を目的として適用され，組み込みシステムが要求するフェイルセーフ機能やフェイル・オペラティブ機能を実現するた

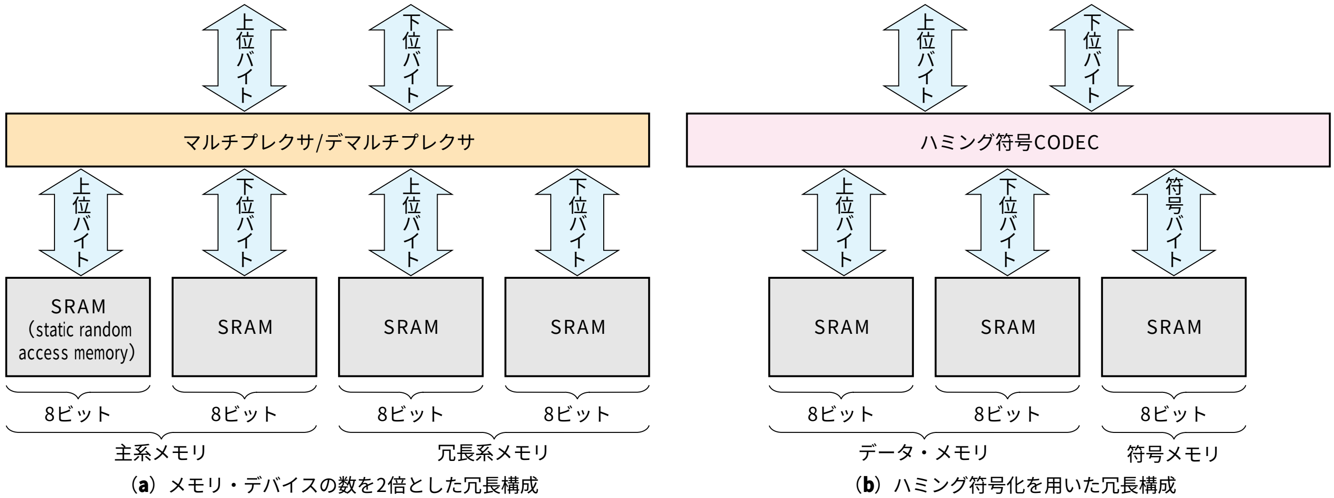


図4　メモリ・システムを冗長化する手段
符号化技術を用いることで単純に複数のメモリを使う構成よりもメモリの数を減らせる．

めの重要な手段となります．

冗長設計を構成するには，デバイス・レベルで冗長を構成するものと，マイクロプロセッサやプロセッサ・モジュール・レベルで冗長を構成するものがあります．ハードウェアの機能で冗長化するばかりでなく，ソフトウェアを用いて冗長化する方法もあります．

● デバイス・レベルの冗長構成は大きく三つ

デバイス・レベルの冗長構成としては図3に示すように，メモリ・システムを冗長化するもの，インターフェース・デバイスを冗長化するもの，および電源を冗長化するものなどがあります．

1）メモリ・システム

メモリ・システムを冗長化する手段としては，パリティ・ビットを追加したりチェック・サム・データを追加したりしてデータ・エラーを検出し，エラーが発見された場合にはデータを書き直したり，あるいは別のメモリを使ったりするなどの方法があります．符号化技術を用いることで単純に複数のメモリを使うよりも冗長なメモリの数を減らすことも可能になります．

ハミング符号を使用した場合の例を図4(b)に示します．この方式では，データ・メモリの半分のビット幅を符号として加えることにより，1ビットのエラー検出/訂正が可能となるため，メモリの個数を減らすことができます．もちろん，エラーが発見された場合には，システムの空き時間などを活用して正しいデータに書き戻しておき，エラーが蓄積して2ビット以上のエラーが発生しないようにします．

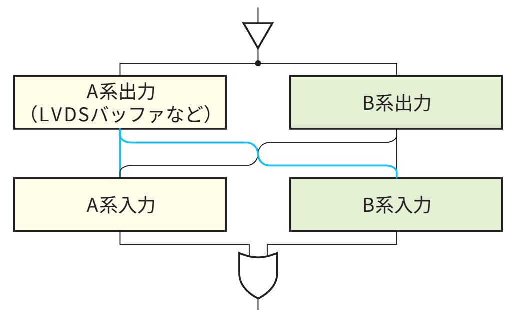


図5　インターフェース・デバイスの冗長化を図るクロス・ストラップ接続
デバイスの故障ばかりでなく，コネクタの接触不良や計装配線の断線などが生じても機器の運用を継続できる．

2）インターフェース・デバイス

インターフェース・デバイスの冗長化を図るものとしては，クロス・ストラップ（たすきがけ）接続がよく使われます．図5のようにインターフェース・デバイスを接続することにより，デバイスの故障ばかりでなく，コネクタの接触不良や計装配線の断線などが生じても機器の運用を継続できます．冗長系の呼称としてはA系，B系と呼ぶことにより，チャネル番号との混同を防ぐなどの配慮をします．

3）電源

電源の冗長化を図る場合には，電源断の状態の予備機を用意しておき，ある電源が落ちた場合には，いったんシステム運用を中断して，予備機の電源に交換するコールド・スタンバイ方式や，あらかじめ複数の電源を並列に接続するホット・スタンバイ方式があります．いずれの方式を用いるにしても，ダイオードやスイッチなどで適切に分離し，電流が逆流しないように注意する必要があります．これら

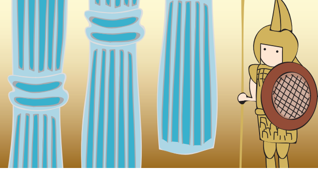


表1 機能分散化要求への対応例

	演算機能の分散化	検出機能の分散化	故障分離・再構成機能の分散化	再同期機能の分散化
メモリ・システム：ECC	○	△	△	―
パリティ	○	△	―	―
チェック・サム	○	△	―	―
インターフェース・デバイス：クロス・ストラップ（たすき掛け）接続	○	―	―	―
電源：コールド・スタンバイ	○	△	―	―
電源：ホット・スタンバイ	○	△	△	△

○：対応可能，△：ほぼ対応可能

のデバイス・レベルの冗長構成をどこまで採用するかについてのトレードオフの例を**表1**に示します．それぞれの機能の分散化要求に対し，どこまで対応するかに応じて各デバイス・レベルで冗長化するかしないかを判断することになります．

● 宇宙機に見るプロセッサ・レベルの冗長構成例

マイクロプロセッサやプロセッサ・モジュール・レベルの冗長構成としては，コールド・スタンバイ方式（待機冗長方式）とホット・スタンバイ方式（常用冗長方式），およびそれらの混在構成があります．デバイス・レベルの冗長化と比べると，比較的大きなモジュール単位での冗長構成が採られることから，さまざまなアイデアが実用化されています．

ここでは，冗長構成が採られることの多い，宇宙機器のプロセッサ・モジュールの例を見てみましょう(1)．宇宙機器は，地球大気圏を越える遠距離で運用され，搭載される組み込み計算機が修理できるケースがまれであることから，冗長構成を採ることにより所定の運用期間を確保する対策が採られます．宇宙機器におけるプロセッサ・モジュール（CPUやPEと呼ばれる）の多重化の目的は，単一点故障[注2]を防止することです．つまり，一部分の異常・故障によりシステム全体の機能が失われることを防止することを目的とします．もちろん，冗長構成だけで高信頼性システムが構築できるわけではないので，各冗長構成方式には得失や課題があり得ます．従って，上述したデバイス・レベルの冗長構成の検討と同じように，どの程度の冗長設計を採用するかを信頼性要求や安全性要求に基づいて検討することになります．

注2：単一故障点は，ハードウェアまたはソフトウェアにおいて，そこが壊れるとシステムの機能が遂行できなくなる個所を指す．単一点故障は，1カ所の故障でシステムの機能が遂行できなくなる現象という意味で使い分けている．

5 コールド・スタンバイ方式とホット・スタンバイ方式

ここでは，前項で触れたコールド・スタンバイ方式（待機冗長方式）とホット・スタンバイ方式（常用冗長方式）について，詳しく説明します．

● コールド・スタンバイ方式：通常は1系統で運用

コールド・スタンバイ方式は，通常は1系統だけ電源を供給して運用しており，異常や故障が生じた場合には他系統に電源を入れ，切り替えて運用を再開するものです．

この方式は天文観測などを目的とした科学衛星の姿勢軌道制御装置に早くから用いられています．2006年初めに打ち上げられた「あかり」(12)などの科学衛星は，カメラやアンテナを地球も含めた各種天体に向けるために，フレキシブルな姿勢・軌道制御を行います．これら科学衛星に搭載されている姿勢・軌道制御装置は，非常にインテリジェント化が進んでおり，冗長構成としては簡潔なコールド・スタンバイ方式を採ることにより，システムの複雑さのバランスを取っています．

コールド・スタンバイ方式はその性格上，主系から冗長系への切り替えに時間がかかるため，主としてフェイルセーフ機能に重点が置かれたものです．

● ホット・スタンバイ方式：常に複数系統で運用する

ホット・スタンバイ方式は，冗長系として用意した複数のモジュールに同時に電源を入れておいて運用するものです．この方式には以下のようなものが挙げられます．各方式の特徴や課題についても述べます．

1）多数決方式

多数決方式は，各CPUの入出力に多数決回路を設け，一

つのCPUが異常・故障を起こしても，全体的には停止しないことを目的としています．一般に高い信頼性が得られますが，この方式は多数決回路そのものの異常や故障には対応できません．つまり，CPUは多重化されているものの，通常は多数決回路で信号が1点に集中しており，多数決回路そのものの異常や故障はシステム・ダウンにつながります．言い換えれば，本構成では多数決回路そのものが単一故障点となっています．このため，多数決回路として高信頼性を持つ特殊なデバイスを用いることで対処しており，デバイスの故障率に依存した確率的な故障率低下と，多数決構成による機能的な解決策とを併用したものとなります．

多数決回路そのものを多重化する対応策もありますが，この場合でも入出力が一つに集まる点があり，システム的にも重量，消費電力などが増大します．多重度を増すにあたり，例えば3重多数決を4重多数決に拡張する場合は，多数決回路周りを新規に設計する必要があります．

この方式はもっとも早くから検討されており，使用実績も多く，さまざまな改良が施され，上記の欠点を補うような対策も種々発表されています．代表的なシステムとしては，平成6年(1994年)度に打ち上げられた技術試験衛星VI型(ETS-VI)に搭載された姿勢制御装置が挙げられます[(3)]．このシステムでは，システム・バスの2重化，記憶部の3重多数決化などが用いられ，信頼度の高いシステムが構成されています．また，姿勢制御装置に対する要求を考慮することにより，多数決回路に起因する単一点故障をシステム的に排除する構成を実現しています．ETS-VI搭載用ACE(姿勢制御電子回路)内に実装された計算機は，ETS-VIの頭脳とも言えるもので，16ビットのCPUモジュールを4台内蔵しています．各CPUモジュールにはそれぞれ3重多数決ローカルRAMを持ち，システム・バス，共有メモリ・モジュール，出力ポートなども2重化されており，さらに共有メモリ・モジュール内のRAMも3重多数決化されています．

一方，システム・バス・インターフェースについては，二つのCPUの並列運転(デュプレックス)方式が採られています．計算機部をフォールト・トレラント化する際に，設計のトレードオフとして，デュプレックス方式と3重多数決方式との比較が行われていますが，姿勢制御において数サイクルの制御の中断は許容可能なため，信頼度・消費電力の観点からデュプレックス方式が採用されたと報告されています．この例は大規模なシステムですが，多数決回路に起因する単一点故障を回避するとともに，システム要求の範囲内においてフェイル・オペラティブを実現し得るシステムと言えます．

2）ソフトウェア通信方式

ソフトウェア通信方式は，各CPUが基本的には並行して処理を行い，あらかじめ決められたチェック・ポイントにおいて互いに通信を行い，データを照合しながら処理を進めていくものです．この方式は，異常や故障が起こった場合，故障の分離，システムの再構成および再同期(正常なマイコンにマスタ機能を移し，ロールバック処理などを行ってシステムの動作を続行する)が比較的容易にできます．

組み込まれるソフトウェアは常に通信し，結果を照合しなければならず，そのため本方式が適用されるシステムは，比較的大規模なシステムとなります．また，複数のCPUをシステム・バスに接続する場合，やはり多数決回路などによって信号を一つにまとめる必要があり，このハードウェアそのものは本方式にとっては本質的でないにもかかわらず，単一故障点を内在せざるを得ないジレンマを持っています．一方で，本方式はソフトウェアによる通信を利用することから，既設計の計算機を流用することが可能になります．スペース・シャトルに用いられているガイダンス・コンピュータは，この特徴をうまく利用しています[(4)]．

日本で最初に月に到達した科学衛星「ひてん」に搭載されたOBCがこの方式を用いています[(5)]．このOBCはデバイスのハイブリッド化を行い，小型化を達成しています．冗長構成としては比較的大規模になりやすいソフトウェア通信方式を使いながら，高度なデバイス技術により小型化した好例と言えます．

3）FRM(functional redundancy monitor)**方式**

FRM方式は，CPUに自己のバス(アドレス，データ，ステータス)出力と外部のバス信号の内容とが一致している/いないの比較監視機能を持たせるものです[(6)]．各CPUは監視モードまたは通常モードといった走行モードを指定できます．

通常モード走行のプロセッサは，この比較回路を動作させずチップ内の処理結果をそのままバスに出力します．監視モードのプロセッサは，チップ内の処理結果をバスに出力せず比較回路の入力とし，また，外部からの信号を比較回路の入力とします．従って，各CPUのデータ・バス，アドレス・バス，ステータス信号線は，おのおの単純にピ

1

ン・ツー・ピン接続するだけで，プロセッサ間でお互いの処理内容を監視できます．比較回路が不一致を検出すると，監視モードのプロセッサは外部回路に対して専用端子を通してこれを通知します．この方式は，多数決方式やソフトウェア通信方式に対して，以下のような利点があります．

- 比較・検出回路が各CPUにそれぞれ内蔵されているため，多数決方式と異なり，これらの回路が単一故障点とならない．
- ソフトウェア通信方式と異なり，常に信号の比較が行われる．また，CPU上のソフトウェアは，比較が行われていることを通常は意識する必要がない．

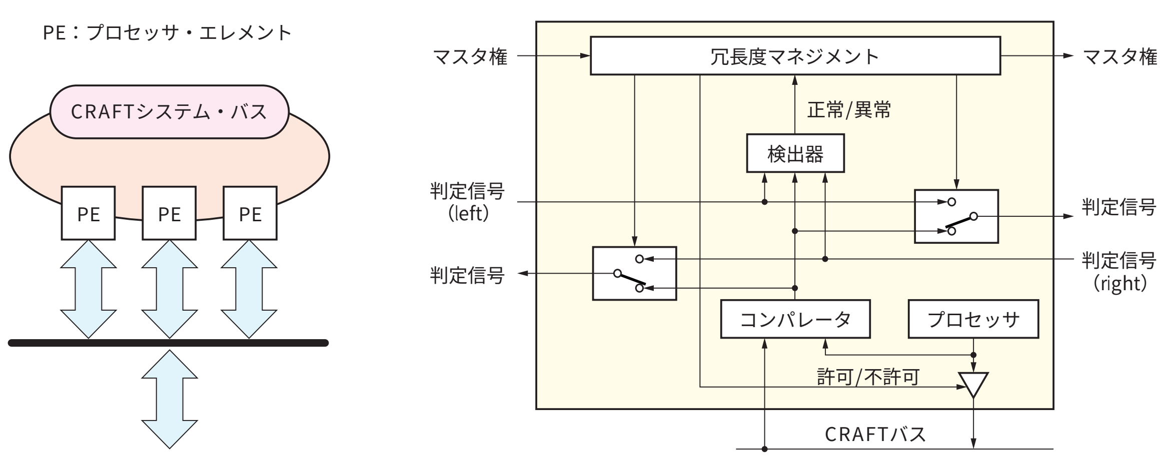
一方，主な課題として次のものが挙げられます．

- マイコン内蔵の機能であるため，使うマイコンが限定される．国産のマイコンでこの機能を有するものは，NEC製のV70など．
- 再構成を行うためには，多数決回路相当のものが必要．具体的には，N重系を構成する各CPUが出力する異常検出信号をまとめて判断し，正常なCPUを選択した上でバス上に信号を出力する権利を持つCPUを指定する機能が必要となる．なぜなら，各CPUが誤って異常検出信号を出力してくる可能性があるため．従って，この異常検出信号を取りまとめて判断する部分は，構成法によっては多数決回路のように単一故障点となり得る．この方式を用いた代表的な装置としては，宇宙ステーション日本実験モジュール（きぼう）に搭載される通信制御装置（JCP）が挙げられる[7]．

4）CRAFTシステム（compare and rational abort for fault-tolerant）

CRAFTシステムは，搭載したコンピュータが持つ各機能（演算機能や故障検出機能，故障分離/再構成機能，再同期機能）の分散化をさらに進めることによって，単一点故障を回避するものです．この方式は以下の目標のもとに開発されたもので，日本独自の方式です[8][9]．

- 原理的に単一故障点を持たないこと．つまり，一つのデバイスの故障によって，システム・ダウンにつながらないこと．
- どのようなデバイスにも適用可能であること．マイコンの世代交代は速い．特定のマイコンに依存した方式では，マイコンの寿命とともに多重化方式の寿命も尽きてしまう．また，汎用マイコンばかりでなく，ディジタル・シグナル・プロセッサなどの特殊用途プロセッサや，周辺素子などにも拡大して適用できることが望ましい．
- 多重度を容易に高められること．例えば，3重冗長システムから4重冗長システムへは，ハードウェアの再設計をほとんど必要とせずに拡張できることが望ましい．
- 基本的にソフトウェアは多重化されていることを意識せずに済むこと．組み込みシステムに搭載されるソフトウェアは年々大規模になってきている．ソフトウェアを開発する場合に，なるべく負荷を少なくする必要がある．
- システムがリアルタイムで再構成されること．バスの制御権を持つCPUに異常が発見された場合，バス・マスタの交代のためにシステム動作を中断する時間を極力なくすものとする．



（a）計算機のモジュール構成　（b）（a）に示したPEの内部

図6　CRAFTシステム

搭載したコンピュータが持つ各機能の分散化をさらに進めることによって，単一点故障を回避する．

図6(a) はSERVIS-2に搭載されている計算機[8]のモジュール構成であり，各CPUモジュールは異常検出ネットワークを構成しています．**図6(b)** はそのうちの一つのCPUモジュールの詳細図です．

各CPUモジュールは，出力比較信号線を通じてリング状に接続され，異常検出ネットワークを構成しており，各CPUモジュール内部に設けられた比較回路の比較結果と，両隣に接続されたCPUモジュールから出力される比較結果との合計三つの信号から，自CPUモジュールが正常か異常かを判断する検出回路を持っています．

各検出回路は自CPUモジュールが異常と判断した場合，CPUモジュール自体をネットワークから論理的に切り離し，マスタ権を隣接したCPUモジュールに移管し，自モジュールの比較結果と隣接CPUモジュールからの比較結果入力とを切り替えて(つまりバイパスして)，隣接CPUモジュールに出力する回路を持っています．

異常が発生したCPUモジュールはシステムから隔離され，n個のCPUモジュールから構成されていたシステムから$n-1$個のCPUモジュールで構成される縮退システムへの再構成が瞬時に行われます．

マスタCPUに異常が発生した場合は，隣接CPUモジュールが制御権を得て，マスタとして機能するようになります．本システムでは異常出力はマスクされ，常に正常な出力が期待できます．この方式では3モジュール以上であれば理論的にn重($n \geqq 3$)までの回路の変更なしに拡張可能で，システム冗長性要求の変化に対して容易に対応できます．また対象となるモジュールがCPUである必要はなく，DSPやI/Oプロセッサなど，ほかの機能デバイスにも応用可能です．

＊　＊　＊

これらの冗長設計を行う際は，十分に故障ケースの解析を行い，一過性の故障に対応するのか，永久故障に対応するのかなども考慮して最適な方式を選択する必要があります．上述の各冗長構成の開発トレンドを**図7**に示します．故障分離や再構成機能の分散化までは冗長方式によって対処可能ですが，再同期処理の分散化は，システム的アプローチも併用されることになるものと考えられます．このようなトレードオフの検討を進めるために，近年，冗長システムを構成するのに適した組み込みシステム用通信ネットワーク規格が提唱されています[10]．評価用システムも発売されており，冗長構成のさまざまな実験ができるようになっています[11]．

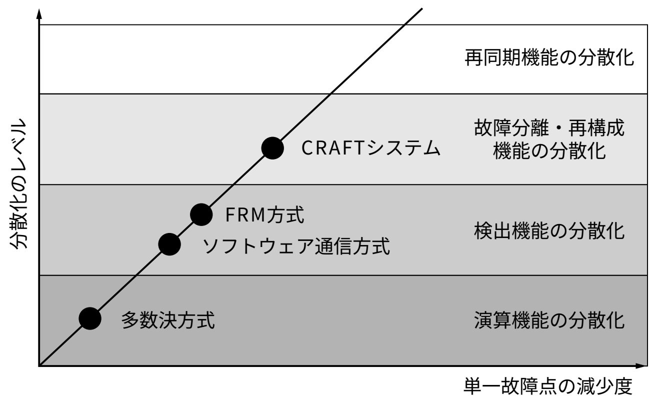


図7　各冗長構成の開発トレンド
故障分離や再構成機能の分散化までは冗長方式によって対処可能だが，再同期処理の分散化は，システム的アプローチも併用されることになる．

参考・引用*文献

(1) 檜原弘樹，大塚誠，水島泰彦，森里司，大島武，三好弘晃，馬場研一；宇宙機搭載用フォールトトレラントコンピュータ，情報処理，vol.35，no.6，1994年．
(2) はやぶさ，宇宙航空研究開発機構宇宙科学研究本部．http://www.isas.ac.jp/j/enterp/missions/hayabusa/
(3) 川田恭裕，石毛康夫；人工衛星姿勢制御用コンピュータシステム，電子情報通信学会誌，vol.73，no.11，pp.1215-1221，1990年．
(4) SklarOFF，J.R.；Redundancy Management Technique for Space Shuttle Computers，IBM J.R&D，20，1，p.20，1976．
(5) 金川信康，井原廣一；宇宙機搭載用コンピューター汎用LSIによるシステム構成ー，電子情報通信学会誌，vol.73，no.11，pp1209-1214，1990年．
(6) 古城 隆，矢野陽一；汎用マイクロプロセッサチップ，電子情報通信学会誌，vol.73，no.11，pp.1222-1227，1990年．
(7) Yamazaki,K.，Hosaka,K.，Morita,N.，Ishii,S.，Kikuchi,T.，and Nakamura,T.：Development Model of Main Processing Controller for Japanese Experiment Module，NEC R&D，vol.34，no.3，pp.385-395，1993．
(8) 檜原弘樹，山田一幸，安達昌紀，三谷健司，秋山雅胤，濱一守；SERVIS-2衛星搭載用自律フォールトトレラント計算機，電子情報通信学会技術研究報告，DC2002-75，pp.19-24，2002年12月．
(9) 特許第2035424号(フォールト・トレラント・コンピュータ・システムやCRAFTシステムの原理が述べられている)．
(10) スペースワイヤ，宇宙科学研究本部．http://www.astro.isas.jaxa.jp/~takahasi/SpaceWire/index_SpaceWire.html
(11) SEMC5701B，シマフジ電機．http://www.shimafuji.co.jp/product/spacecube01.html
(12) AKARI，宇宙航空研究開発機構宇宙科学研究本部．http://www.isas.ac.jp/j/enterp/missions/akari/

ひはら・ひろき
組込みソフトウェア管理者・技術者育成研究会(SESSAME)

1